CounterACT™ Macintosh/Linux Property Scanner Plugin

Configuration Guide

Version 7.0.1 and Above

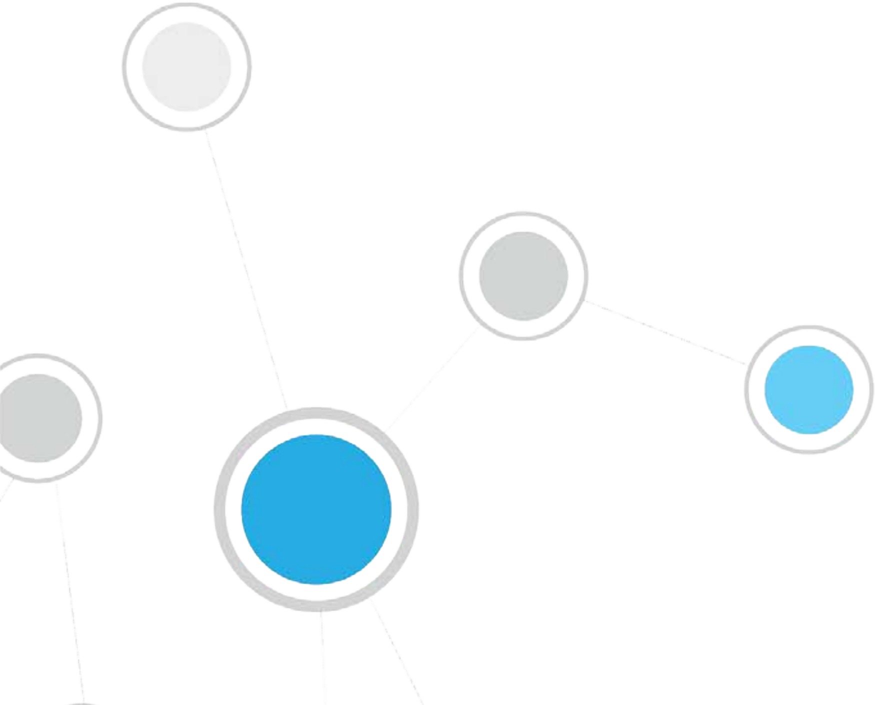


Table of Contents

About the Macintosh/Linux Property Scanner Plugin

The Macintosh/Linux Property Scanner Plugin supports the following functionality on Macintosh and Linux endpoints that CounterACT detects in your environment:

- Access Macintosh and Linux endpoints.
- Perform comprehensive, deep inspection for the purpose of resolving an extensive range of endpoint information, such as operating system details, applications installed, file information and more.
- Remediate endpoints by using Macintosh and Linux CounterACT actions. For example, run scripts or initiate Macintosh updates on endpoints.

This document describes how to configure the plugin and provides other information including supported operating systems, executables and processes generated by the plugin, and troubleshooting issues.

> 🗎 *The HPS Inspection Engine Plugin provides this functionality for Windows endpoints in your environment. See the HPS Inspection Engine Plugin Configuration Guide for details.*

Plugin Updates

The plugin is bundled with major CounterACT releases. New releases may become available between major version releases. The **Plugin Updates** icon appears on the status bar of the Console when a plugin update is available.

Requirements

The following CounterACT components must be installed to use this release.

- CounterACT version 7.0.0
- CounterACT Service Pack 2.0.1 or above
- HPS Inspection Engine Plugin version 10.2.5.1 or above

Supported Operating Systems

This section describes the endpoint operating systems supported by this release.

- **Macintosh:**
  - Mac OS X versions 10.6 through 10.11
  - The following shell types running on Macintosh endpoints: *sh*, *bash*, *csh* and *tcsh, zsh.*

  > 🗎 *The new OS X Plugin now supports SecureConnector for endpoints running version 10.8 and above of the OS X operating system. It is strongly recommended to install the OS X Plugin when you upgrade to this release. See Appendix 3: Migrating Macintosh Endpoints Managed by SecureConnector to the OS X Plugin for details.*

- **Linux:**
  - CentOS version 5 and above

- – Debian version 8 and above
- – Fedora version 18 and above
- – Red Hat Enterprise Linux version 5 and above
- – Red Hat Enterprise Linux Desktop version 7 and above
- – Red Hat version 7.2 and above
- – OpenSUSE version 12 and above
- – SUSE Enterprise version 11 and above
- – Ubuntu version 12.04 and above

Accessing and Managing Endpoints

The plugin accesses endpoints to learn detailed information such as file meta-data, operating system information, and more. In addition, the plugin is used to run scripts on endpoints and to perform other remediation actions.

When you configure the plugin, you determine the methods you want to use to access and manage endpoints. When CounterACT successfully implements these access methods on an endpoint, the endpoint is resolved as *Manageable* by CounterACT.

The plugin provides the following methods to access endpoints:

- Remote Inspection (for Macintosh and Linux endpoints)
- SecureConnector (for Linux endpoints only)

Both methods can be deployed together in a single network environment.

Remote Inspection

Remote Inspection uses the SSH communications protocol to query the endpoint and to run scripts and implement remediation actions on the endpoint.

Agentless

Remote Inspection is *agentless* - CounterACT does not install any applications on the endpoint to query it. This makes Remote Inspection useful when administrators or end users do not want to install utilities or other executables on the endpoint.

Specify remote inspection settings in the Remote Inspection tab during plugin configuration. For more information, see What to Do and Managing Endpoints Using Remote Inspection.

SecureConnector

SecureConnector is a small-footprint executable that runs on the endpoint. It reports endpoint information to CounterACT, and implements actions on the endpoint. The *Start SecureConnector* action installs SecureConnector on endpoints.

> 🗎 *The new OS X Plugin now supports SecureConnector for endpoints running the OS X operating system. The Macintosh/Linux Property Scanner Plugin supports existing Macintosh endpoints using the legacy version of SecureConnector installed on the endpoint, but you must install the OS X Plugin to update SecureConnector on OS X devices. See Appendix 3:*

Migrating Macintosh Endpoints Managed by SecureConnector to the OS X Plugin for details.

Agent-Based

The SecureConnector executable file must be installed and maintained on the endpoint. This may not be acceptable in certain network environments, or for some endpoints or users. SecureConnector can be installed in several ways:

- As a dissolvable utility
- As a permanent application
- As a permanent service; see Appendix 1: Using Linux Packages to Deploy SecureConnector.

For more information, see What to Do and Managing Linux Endpoints Using SecureConnector.

What to Do

1. Configure the Plugin
2. To work with Remote Inspection on Linux or OS X endpoints:
   - **a.** Define settings in the Remote Inspection tab of the Plugin Configuration pane.
   - **b.** Distribute the Public Key to endpoints.
   - **c.** Define a Remote Inspection User on Macintosh and Linux Endpoints.
3. To work with SecureConnector on Linux endpoints: Include the **Start SecureConnector** action in compliance policies. The action installs the SecureConnector executable on endpoints detected by the policies.

   To work with SecureConnector on OS X endpoints: from this release, the Macintosh/Linux Property Scanner Plugin no longer supports SecureConnector interaction with OS X endpoints. To maintain management continuity with existing OS X endpoints using legacy versions of SecureConnector, see Appendix 3: Migrating Macintosh Endpoints Managed by SecureConnector to the OS X Plugin.

4. Test the Plugin
5. Create a policy to Make Macintosh/Linux Endpoints Manageable.

Configure the Plugin

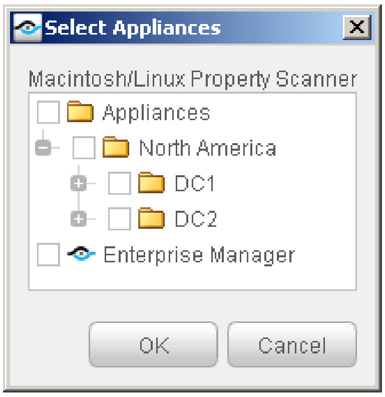
Configure the plugin to:

- Define global settings for Remote Inspection
- Define how the plugin identifies user accounts on endpoints
- Specify test parameters and test connectivity

To configure the plugin:

1. In the CounterACT Console, select **Options** from the **Tools** menu.

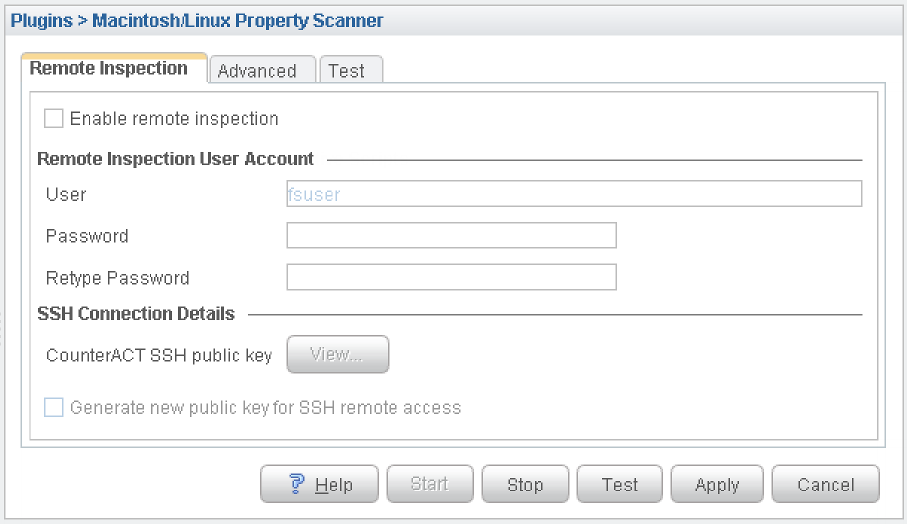
2. Select **Plugins**. In the Plugins pane, select the Macintosh/Linux Property Scanner Plugin. Select **Configure**. The Select Appliances dialog appears.



3. Do one of the following:
   – Select the Enterprise Manager and select **OK**. Continue with Step 4.
   – Select an Appliance and select **OK**. Continue with Step 6.

🗎 *Select Enterprise Manager for initial configuration of the plugin. When you select Enterprise Manager, all configuration options are available. When you select an Appliance, only a limited subset of configuration settings is available.*

4. The Remote Inspection tab is displayed.



□ Enable remote inspection

Remote Inspection User Account

User fsuser

Password

Retype Password

SSH Connection Details

CounterACT SSH public key View...

□ Generate new public key for SSH remote access

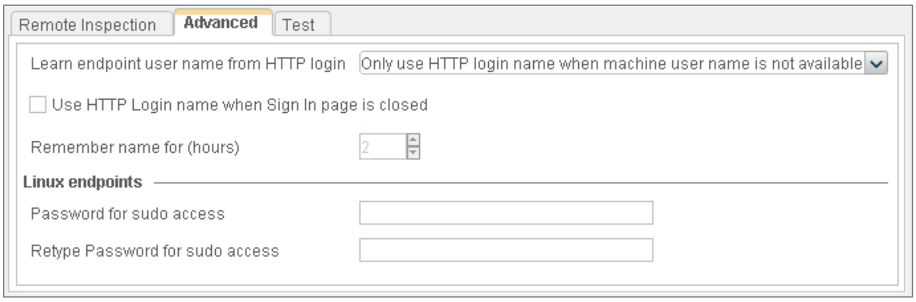
5. Configure the following options.

🗎 *When you configure the Enterprise Manager, these settings are copied to all Appliances in the network.*

Enable remote inspection	Select this option to enable use of Remote Inspection methods to poll endpoints for information. The other fields of this tab are only relevant if Remote Inspection is used in your environment.
Remote Inspection User	Specify an administrator user account that is used to establish an SSH connection with endpoints. This user account must be defined on each Macintosh/Linux endpoint.

Account	A valid password must be provided to use actions or properties that require privileged access, such as the **Macintosh Software Updates Missing** property or the *Run Interactive* option of the **Run Script on Macintosh** action.
CounterACT SSH public key	Select **View** to see the public key CounterACT uses for the SSH connection to endpoints. This key must be distributed to endpoints. See Distribute the Public Key for details.
Generate new public key for remote SSH access	For increased security, select this option and select **Apply** to change the public key. The plugin changes the public key of the Enterprise manager, and synchronizes all Appliances with the new key. You must distribute the new key to endpoints using one of the methods described in Distribute the Public Key.

6. Select the Advanced tab.



Learn endpoint user name from HTTP login: Only use HTTP login name when machine user name is not available

□ Use HTTP Login name when Sign In page is closed

Remember name for (hours): 2

Password for sudo access

Retype Password for sudo access

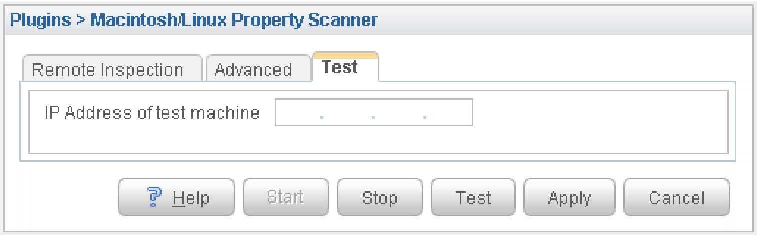
a. Configure the following options. When you configure the Enterprise Manager, these settings are *not* copied to all Appliances in the network. You can configure these settings differently for each Appliance.

Learn endpoint user name from HTTP login	Indicates the method used for learning endpoint user names. This information is used to evaluate the **User** host property.
Use HTTP Login name when the Sign In page is closed	Unless a new user login occurs, the User host property retains the username of the most recent HTTP login session, even after the session is closed.
Remember name for (hours)	Indicates how long the plugin retains the HTTP login name when the sign in page is closed. This time is calculated from the last successful login.

b. (Optional) Configure the following option. This setting is applied uniformly to all Appliances in the network.

Linux endpoints: Password for sudo access	The plugin uses the sudo utility when super user access is required to run scripts on Linux endpoints. For example, sudo mode is invoked when the *Run script as root user on endpoint* option is enabled for the **Run Script on Linux** action or the **Linux Expected Script Result** host property. To use this feature, configure Linux endpoints in your environment to use a fixed password for sudo access, and use this field to specify this password. For example, you can specify the root password in this field, and add the following line to the /etc/sudoers file: `Defaults rootpw` On endpoints where sudo mode is not password protected, this field is ignored.

7. Select the Test tab.



Plugins > Macintosh/Linux Property Scanner

IP Address of test machine

8. Enter the IP address of an endpoint that is used to test the plugin's ability to connect to Macintosh/Linux endpoints.

   🗎 *When you configure the Enterprise Manager, this setting is* not *copied to all Appliances in the network. It is recommended to test endpoint connectivity from each Appliance that manages Macintosh/Linux endpoints. Configure each Appliance and specify an IP address within the scope of the Appliance.*

9. Select **Apply** to save settings.

   🗎 *When you configure the Enterprise Manager, settings of the Remote Inspection tab are copied to all Appliances in the network.*

10. Select **Start** to start the plugin with these settings.

11. Repeat this procedure to configure settings of the Advanced and Test tabs for all Appliances that manage Macintosh/Linux endpoints.

Test the Plugin

You can verify that the plugin is configured correctly and can connect to Macintosh/Linux endpoints.

To test the plugin:

1. Verify that test endpoints were defined during plugin configuration.
2. If you are working with Remote Inspection, verify that the following steps were completed on all test endpoints:
   - The Remote Inspection user defined during plugin configuration exists.
   - The public key used by CounterACT was installed.
3. In the CounterACT Console, select **Options** from the **Tools** menu.
4. Select **Plugins**. In the Plugins pane, select the Macintosh/Linux Property Scanner plugin. Do one of the following:
   - To test connection to all test endpoints, select **Test**.
   - To test connection to the test endpoint defined for a specific Appliance:

     Select **Appliances**.

     In the Appliances Installed dialog, select an Appliance and select **Test**.
5. Select **Yes** to continue with the test.

Make Macintosh/Linux Endpoints Manageable

The standard Asset Classification policies provided with CounterACT identify Macintosh and Linux endpoints, and assign these endpoints to the *Macintosh* or *Linux/Unix* group. Create a policy that uses the **Linux Manageable** and **Macintosh Manageable** host properties to detect members of these groups that are not yet manageable by CounterACT.

- To make an endpoint manageable by Remote Inspection, a policy rule should do the following on the endpoint:
  - Define a Remote Inspection User on Macintosh and Linux Endpoints
  - Distribute the Public Key
- To make a Linux endpoint manageable by SecureConnector, a policy rule should Install SecureConnector on the endpoint.

Managing Endpoints Using Remote Inspection

You can inspect endpoints using SSH remote access. SSH remote access is implemented by distributing the Appliance's public key to managed endpoints.

You may need to disable remote inspection if it causes too many unnecessary access attempts by CounterACT at the endpoints. If you disable remote inspection, you can use SecureConnector to manage devices. See Managing Linux Endpoints Using SecureConnector for information about SecureConnector setup.

Define a Remote Inspection User on Macintosh and Linux Endpoints

Define an admin-level user on each endpoint that you want to manage. This user should have the name you entered in the **User** field of the Remote Inspection tab during plugin configuration.

Distribute the Public Key

The public key allows SSH-based inspection of the endpoint without the endpoint user's password. You may need an endpoint password to distribute the key. Two options are available for key distribution:

- Run a predefined ForeScout command that distributes the key. See fstool pkdis – Distribute a Public Key
- Create a Custom Script that Distributes the Key

fstool pkdis – Distribute a Public Key

You can use the ForeScout fstool pkdis utility to distribute the key to a single IP address, a range of IP addresses or a list of endpoint IP addresses.

The first time a key is distributed to an endpoint, you must enter an endpoint user name and password. No password is needed for subsequent updates of the public key.

If you use a list or a range of endpoint addresses, the same administrator user account must be defined on all the endpoints in the list/range, with the identical user name and password. See Define a Remote Inspection User on Macintosh and Linux Endpoints.

To use the fstool pkdis utility to distribute the public key:

1. Log on to an Appliance or Enterprise Manager as root.
2. Run the following command:

   fstool pkdis –u <*username*> **–h** <*ip_address*> **|** <*ip_range*> **|** <*ip_list*>

   Where:

   <*username*> is the user you specified during plugin configuration and defined on each endpoint, or an administrator account on the endpoint.

   <*ip_address*> is an IP address in dot notation, for example 10.0.0.11

   <*ip_range*> is a dash-separated IP address range, for example:

   10.0.0.5-10.0.0.100

   <*ip_list*> is a comma-separated IP address list, for example

   10.0.0.6, 10.0.0.10, 10.0.0.29
3. Repeat this procedure if you generate a new key pair by selecting **Generate new public key for remote SSH access** in the Enterprise Manager Configuration dialog box (see Configure the Plugin).

Create a Custom Script that Distributes the Key

You may want to create a custom script that distributes the key, for example if you want to use the script to perform additional tasks while distributing the key.

To create a script:

1. In the CounterACT Console, select **Options** from the **Tools** menu.
2. Select **Plugins**. In the Plugins pane, select the Macintosh/Linux Property Scanner plugin. Select **Configure**. The Select Appliances dialog appears.
3. Select the Enterprise Manager and select **OK**.
4. Select **View** in the **CounterACT SSH Connection Details** area of the Remote Inspection tab.
5. Copy the key to a clipboard or another application.
6. Write a script which does the following on each endpoint you want to manage via Remote Inspection:
   a. Create the folder **.ssh** under the user defined in the **Remote Inspection User** field of the plugin Configuration pane.
   b. Change the **.ssh** folder permissions as follows:

      chmod 755 .ssh (there is a space between **755** and the **.ssh** suffix).
   c. Paste the public key into the file **.ssh/authorized_keys.** Save the file.

d. Change the file **`.ssh/authorized_keys`** permissions as follows:

```
chmod 644 authorized_keys
```

Managing Linux Endpoints Using SecureConnector

SecureConnector is a light footprint executable that runs on the endpoint, enabling access to endpoints that cannot be inspected via SSH. SecureConnector creates an encrypted tunnel between endpoint and Appliance through port 2200.

🗎 *The OS X Plugin now supports SecureConnector for endpoints running the OS X operating system. The Macintosh/Linux Property Scanner Plugin supports existing Macintosh endpoints using the legacy version of SecureConnector installed on the endpoint, but you must install the OS X Plugin to update SecureConnector on OS X devices. See Appendix 3: Migrating Macintosh Endpoints Managed by SecureConnector to the OS X Plugin for details.*

Start SecureConnnector

The **Start SecureConnector** action installs the SecureConnector executable if it is not already present on an endpoint.

Secure Connector can be implemented on the endpoint as a dissolvable executable, a permanent application, or a service.

- A dissolvable executable runs once on installation, and does not run again after the user logs out or the machine is rebooted.
- When installed as a permanent application, SecureConnector will run every time the user logs in, and in some cases as soon as the machine boots.
- To install SecureConnector as a permanent service, use the distribution method described in Appendix 1: Using Linux Packages to Deploy SecureConnector .

When the **Start SecureConnector** action is applied to Linux endpoints, configure the following action options as follows:

Install Method	Only the **HTTP installation at the endpoint** installation method is supported. For an alternative method of distributing SecureConnector to Linux endpoints, see Appendix 1: Using Linux Packages to Deploy SecureConnector.
Deployment Type	Only the **Install Dissolvable** and **Install Permanent as Application** options are supported for Linux endpoints. To install SecureConnector as a permanent service on Linux endpoints, use the distribution method described in Appendix 1: Using Linux Packages to Deploy SecureConnector .

Refer to the CounterACT *Console User Guide* and the *HPS Inspection Engine Plugin Configuration Guide* for more information about SecureConnector.

Stop SecureConnector

The **Stop SecureConnector** action uninstalls the SecureConnector executable and deletes it from the endpoint. You can also invoke the following shell command to uninstall SecureConnector:

```
./SecureConnector.sh uninstall
```

SecureConnector Details

Item	Detail
Size on disk	Less than 20kB including setup files on Macintosh and Linux where sshd is installed. Slightly more than 1MB on Ubuntu desktop edition where sshd is fetched from CounterACT during setup.
Memory utilization	1.5 MB for whole suite.
Installation type	Permanent or dissolvable. Defined in the Start SecureConnector action.
Visibility options (systray icon)	Visible and non-visible.
Deployment options	Via browser: HTTP redirection or direct file download
SecureConnector privilege level:	Deployed via HTTP: user privileges RPM or DEB package deployment: root privileges
Permanent installation folder	Any of the following: ▪ The default installation directory, ~/ForeScout/SecureConnector ▪ A CounterACT user-defined directory.
Dissolvable installation folder	/tmp
Permanent script folder	For the permanently installed SecureConnector, the script folder is ~/ForeScout/SecureConnector/SCRIPTS
Dissolvable script folder	/tmp/SCRIPTS
Starts on log in	Permanent mode: Yes Dissolvable mode: No Installation mode is set in the Start SecureConnector action.
Permanent mode installation starts on boot	Linux RPM or DEB package installations can be set to start on boot.

Restrict Usage of SecureConnector by IP Address Range

You can restrict the range of endpoints that can be managed by SecureConnector. This is achieved by allowing only a specified range of IP addresses to connect to the plugin. Endpoints outside this range must be managed by Remote Inspection rather than by SecureConnector, even if endpoint users acquire and install SecureConnector. This restriction supports several common deployment cases, such as:

- Company policy prevents agents on corporate devices.

- Company policy disallows traffic on certain ports, including the SecureConnector communication ports.

To restrict access to SecureConnector by IP address:

1. Log in to the Appliance on which you want to restrict access.
2. Run the following command:

   `fstool mac set_property config.sshd_allow_ranges.value` **<*ip_range*>**

   Where **<***ip_range***>** is a comma-separated list of individual IPv4 addresses or subnet masks. Only IP addresses in this range can contact SecureConnector on this Appliance. The following example limits access to two Class C networks:

   ```
   fstool mac set_property config.sshd_allow_ranges.value
   192.180.100.1/255.255.255.0, 192.185.100.1/24
   ```
3. Restart the Macintosh/Linux Property Scanner Plugin.
4. Repeat this procedure on each Appliance to restrict SecureConnector access to that Appliance.

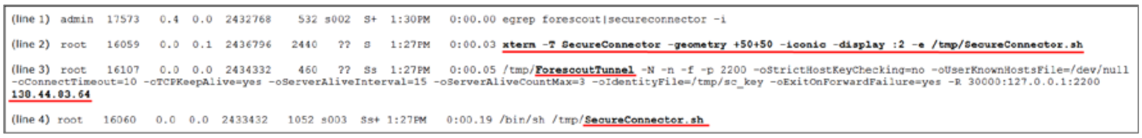
Troubleshooting SecureConnector

If after deploying SecureConnector, the Console shows that particular endpoints are not being managed by SecureConnector, verify that SecureConnector is running on the affected endpoints. Run the following command on the endpoint:

```
ps auxww | egrep 'forescout|secureconnector' -i
```

The resulting output provides you with the following information:

- Identifies that SecureConnector is running by listing the **`SecureConnector.sh`** process. See line 4 in the example below.
- Identifies that the tunnel is active by listing the **`ForescoutTunnel`** process. See line 3 in the example below.
- Identifies that the SecureConnector icon is shown in the user interface of the endpoint, given that the installation option **Show Systray icon** was selected. See line 2 in the example below.



```
(line 1)  admin   17573   0.4  0.0  2432768    532 s002  S+   1:30PM   0:00.00 egrep forescout|secureconnector -i

(line 2)  root    16059   0.0  0.1  2436796   2440   ??  S    1:27PM   0:00.03 xterm -T SecureConnector -geometry +50+50 -iconic -display :2 -e /tmp/SecureConnector.sh

(line 3)  root    16107   0.0  0.0  2434332    460   ??  Ss   1:27PM   0:00.05 /tmp/ForescoutTunnel -N -n -f -p 2200 -oStrictHostKeyChecking=no -oUserKnownHostsFile=/dev/null
-oConnectTimeout=10 -oTCPKeepAlive=yes -oServerAliveInterval=15 -oServerAliveCountMax=3 -oIdentityFile=/tmp/sc_key -oExitOnForwardFailure=yes -R 30000:127.0.0.1:2200
138.44.83.64

(line 4)  root    16060   0.0  0.0  2433432   1052 s003  Ss+  1:27PM   0:00.19 /bin/sh /tmp/SecureConnector.sh
```

Verify that SecureConnector connects to the Appliance that manages the endpoint. See line 3 in the example above; the Appliance IP is **`138.44.83.64.`**

Create Custom Policies

CounterACT *policies* are powerful tools for automated endpoint access control and management.

Properties

Information reported to CounterACT is stored as a *host property*. Host property values are displayed in Console views, and can be evaluated and examined by

CounterACT *policies* to trigger management and remediation *actions*.

Actions

Include CounterACT actions in a policy to remediate or manage detected devices. For example, assign a detected device to a quarantine VLAN or send the device user or IT team an email.

Policies

CounterACT policies are a series of rules. Each rule contains:

- Conditions based on host property values.
- Actions that are applied to endpoints.

When an endpoint matches the conditions of a rule, the actions of the rule are applied to the endpoint.

Creating Policies to Manage Macintosh/Linux Endpoints

This plugin supports host properties and actions specific to Macintosh/Linux endpoints. In addition, the plugin implements the generally applicable actions on Macintosh/Linux endpoints. Use these properties and actions to create policies that automate management of Macintosh/Linux endpoints.

Properties Supported by This Plugin

This section describes the host properties unique to Macintosh and Linux endpoints that are resolved based on information retrieved by this plugin.

Linux Properties

Linux Expected Script Result	Use this property to run a command or file that detects certain endpoint attributes, statuses or any other information defined in the script or command. Commands and scripts can also be used to carry out actions on endpoints. All file extensions are supported and can be run. Select the **Run script as root user on endpoint** option to run the specified script using root user privileges on Linux endpoints. Select this option when a script requires root privileges, but CounterACT does not use root credentials to access the endpoint. To use this option the *sudo* utility must be enabled on Linux endpoints. When sudo mode is password protected, you must configure a password that lets CounterACT enter sudo mode. See Configure the Plugin. The **Run Script Action** is also available. See Run Script on Linux / Run Script on Macintosh.
Linux File Date	Indicates the last time that a file on an endpoint was modified.
Linux File Exists	Indicates the existence of a defined file on an endpoint.
Linux File Size	Indicates the size (in bytes) of a defined file on a Linux device.
Linux Hostname	Indicates the Linux host name.
Linux Manageable (SSH Direct Access)	Indicates whether the endpoint is connected to CounterACT via SSH and is manageable via Remote Inspection.
Linux Manageable (SecureConnector)	Indicates whether the endpoint is connected to CounterACT via SecureConnector.

Linux Processes Running	Indicates the processes currently running on an endpoint.
Linux User	Indicates the user directly logged in to the Linux console.
Linux Version	Indicates the version of the Linux OS running on an endpoint.

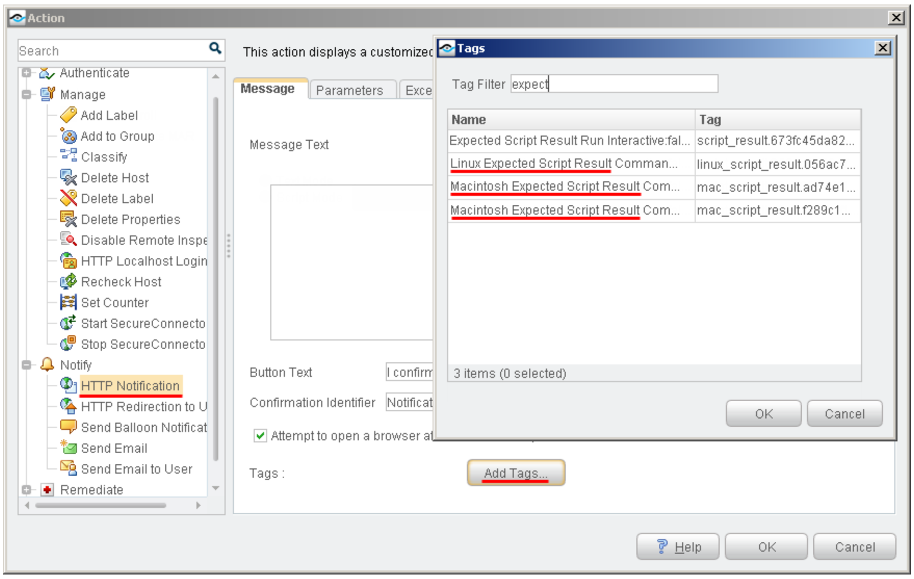
Macintosh Properties

Macintosh Expected Script Result	Use this property to run a command or file that will detect certain endpoint attributes, statuses or any other information defined in the script or command. Commands and file can also be used to carry out actions on endpoints. All file extensions are supported and can be run. The **Run Script Action** is also available. See Run Script on Linux / Run Script on Macintosh.
Macintosh File Date	Indicates the last time that a file on an endpoint was modified.
Macintosh File Exists	Indicates the existence of a defined file on an endpoint.
Macintosh File Size	Indicates the size (in bytes) of a defined file on a Macintosh device.
Macintosh Hostname	Indicates the Macintosh host name.
Macintosh Manageable (SSH Direct Access)	Indicates whether the endpoint is connected to CounterACT via SSH and is manageable via Remote Inspection.
Macintosh Manageable (SecureConnector)	Indicates whether the endpoint is connected to CounterACT via SecureConnector.
Macintosh Processes Running	Indicates the processes currently running on an endpoint.
Macintosh SecureConnector Version	Indicates the version of the SecureConnector package that is running on an endpoint.
Macintosh Software Updates Missing	Indicates Macintosh security and other updates that are missing on an endpoint.
Macintosh User	Indicates the user directly logged in to the Macintosh console.
Macintosh Version	Indicates the version of the Macintosh OS running on an endpoint.

Property Tags for Scripts Results on Macintosh/Linux Endpoints

The **Macintosh Expected Script Result** and **Linux Expected Script Result** properties run scripts on endpoints and evaluate the results. When you create a policy condition using one of these properties, CounterACT generates a property tag that lets you include the script results in action definition fields.

For example, a script returns a list of files in a folder. When you create a condition based on **Expected Script Result** properties that runs the script on each endpoint, CounterACT generates a tag that represents the list of names discovered by the script on the endpoint. Use this property tag in a notification action that sends the end user the list of files found on the endpoint by the script.



Tags

Tag Filter expect

Name	Tag
Expected Script Result Run Interactive:fal...	script_result.673fc45da82...
Linux Expected Script Result Comman...	linux_script_result.056ac7...
Macintosh Expected Script Result Com...	mac_script_result.ad74e1...
Macintosh Expected Script Result Com...	mac_script_result.f289c1...

3 items (0 selected)

In the example shown above, a Console user is editing the message text of an HTTP Notification action. Active policies in CounterACT contain:

- One condition based on the **Expected Script Result** property (for Windows endpoints).
- One condition based on the **Linux Expected Script Result** property.
- ***Two conditions*** based on the **Macintosh Expected Script Result** property.

CounterACT has generated a property tag for each **Expected Script Result** condition. When the user inserts one of these tags in the message, CounterACT replaces the tag with the current value returned by the script on the endpoint.

Note that:

🗎 *The tag contains the result of the script, not the resolved condition value.*

🗎 *The tag is available as long as the condition is present in active policies.*

🗎 *Active policies may contain several unrelated conditions based on an **Expected Script Result** property - each of which runs a different script. CounterACT generates and maintains a* separate *property tag for each instance of the **Expected Script Result** host property.*

Actions Supported by This Plugin

The plugin implements the following actions on Macintosh/Linux endpoints:

- HTTP Login
- HTTP Notification
- HTTP Redirection to URL

In addition, the plugin implements the following actions on Linux endpoints:

- Start/Stop SecureConnector

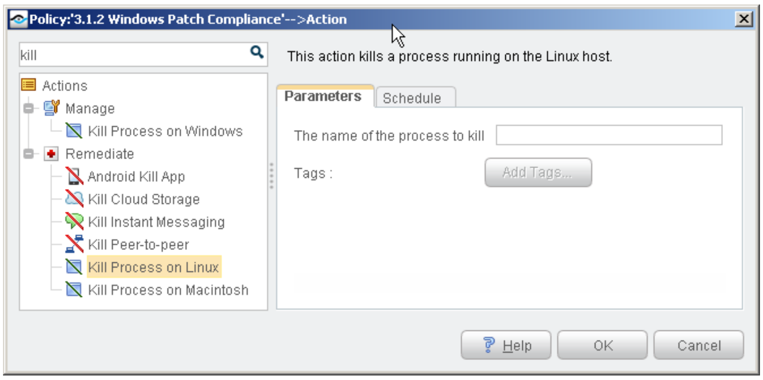
This plugin provides the following actions specific to Macintosh/Linux endpoints:

- Kill Process on Linux / Kill Process on Macintosh

- Migrate to OS X SecureConnector Action
- Run Script on Linux / Run Script on Macintosh
- Start Macintosh Updates

Kill Process on Linux / Kill Process on Macintosh

These actions halt the specified Linux and Macintosh process. You can use property tags to include endpoint-specific or user-specific values in this field. See the *Console User Guide* for details.


Policy:'3.1.2 Windows Patch Compliance'-->Action

This action kills a process running on the Linux host.

The name of the process to kill

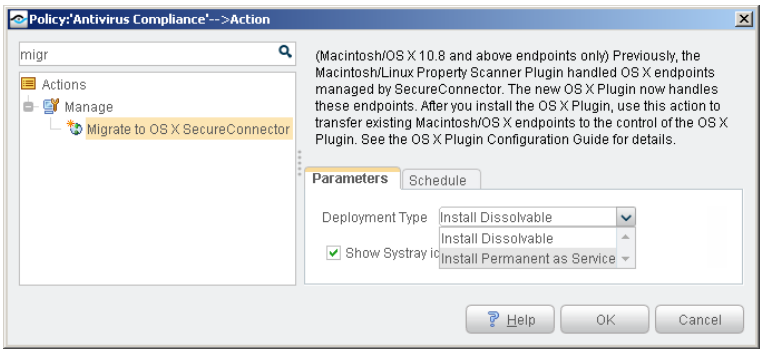
Tags :

Migrate to OS X SecureConnector Action

The OS X Plugin must be installed before you use this action.

This action installs the latest SecureConnector version provided by the OS X Plugin on OS X endpoints running legacy versions of SecureConnector. This version of SecureConnector connects to the OS X Plugin. After migration, the Macintosh/Linux Property Scanner Plugin no longer handles SecureConnector interaction for these endpoints.

See Appendix 3: Migrating Macintosh Endpoints Managed by SecureConnector to the OS X Plugin.


Policy:'Antivirus Compliance'-->Action

(Macintosh/OS X 10.8 and above endpoints only) Previously, the Macintosh/Linux Property Scanner Plugin handled OS X endpoints managed by SecureConnector. The new OS X Plugin now handles these endpoints. After you install the OS X Plugin, use this action to transfer existing Macintosh/OS X endpoints to the control of the OS X Plugin. See the OS X Plugin Configuration Guide for details.

Deployment Type: Install Dissolvable

☑ Show Systray ic

The following options are available for this action:

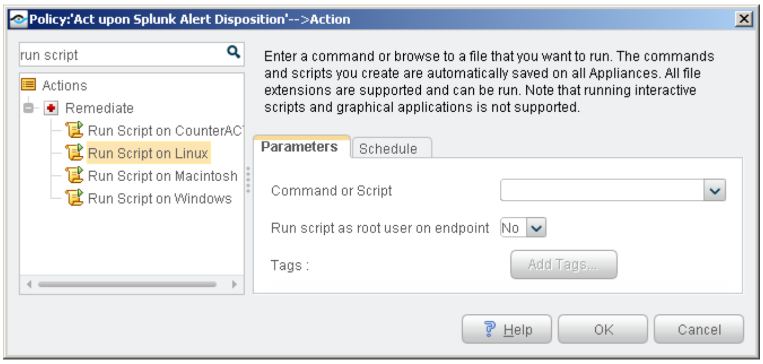
Deployment Type	Determines how SecureConnector is installed on the endpoint. ▪ Install Dissolvable: installs SecureConnector as a dissolvable utility in all open console sessions on the endpoint. ▪ Install Permanent as Service: installs SecureConnector as a Mac OS X Service which runs permanently in the background.

Show Systray Icon	Determines whether an icon for SecureConnector is displayed in the OS X menu bar.

Run Script on Linux / Run Script on Macintosh

You can leverage scripts to:

- Automatically run Macintosh and Linux updates.
- Automatically deploy vulnerability patches and antivirus updates.
- Automatically delete files.
- Create customized scripts to perform any action that you want.

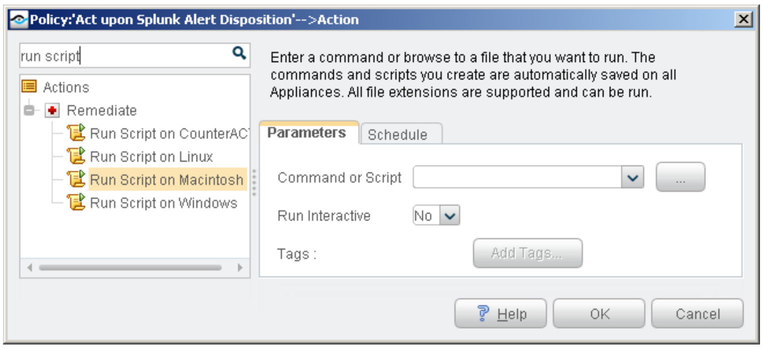


Policy:'Act upon Splunk Alert Disposition'-->Action

Enter a command or browse to a file that you want to run. The commands and scripts you create are automatically saved on all Appliances. All file extensions are supported and can be run. Note that running interactive scripts and graphical applications is not supported.

Command or Script

Run script as root user on endpoint: No



Policy:'Act upon Splunk Alert Disposition'-->Action

Enter a command or browse to a file that you want to run. The commands and scripts you create are automatically saved on all Appliances. All file extensions are supported and can be run.

Command or Script

Run Interactive: No

To use these actions:

1. Specify a command or script to run on endpoints. Do one of the following:
   - – Enter a command in the **Command or Script** field. To run a file that already exists on the endpoint, enter its absolute path. You can use property tags to include endpoint-specific or user-specific values in this field. See the *Console User Guide* for details.
   - – Select the Continue button to select from the repository of user-defined scripts and commands. See the *CounterACT Console User Guide* for more information about user-defined scripts.
2. Specify the following optional behaviors, if required.

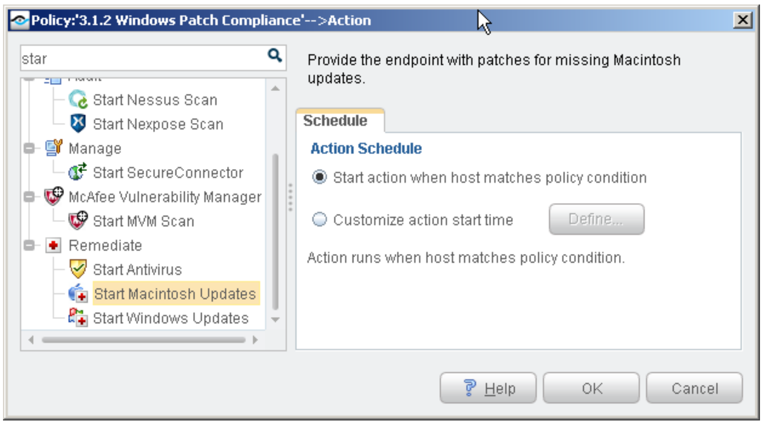
Run interactive (Macintosh endpoints)	Select this option to run the specified command or script interactively on Mac OS X endpoints. On endpoints managed by the OS X Plugin using SecureConnector, prompts are displayed to the currently logged in user in a terminal window. See the *OS X Plugin Configuration Guide* for details.

Run script as root user on endpoint (Linux endpoints)	Select this option to run the specified script using root user privileges on Linux endpoints. Select this option when a script requires root privileges, but CounterACT does not use root credentials to access the endpoint. To use this option the *sudo* utility must be enabled on Linux endpoints. When sudo mode is password protected, you must configure a password that lets CounterACT enter sudo mode. See Configure the Plugin.

3. Use the options of the Schedule tab to specify when the action is applied, to delay application of the action, or to specify repeat application of the action.

Start Macintosh Updates

This action triggers automatic operating system updates. Use the action in policies that use the *Macintosh Software Updates Missing* property to detect endpoints missing software updates.



Policy:'3.1.2 Windows Patch Compliance'-->Action

Provide the endpoint with patches for missing Macintosh updates.

Action Schedule

● Start action when host matches policy condition

○ Customize action start time

Action runs when host matches policy condition.

Appendix 1: Using Linux Packages to Deploy SecureConnector

You can create a Linux package to contain SecureConnector and then distribute the package to Linux endpoints that you want to be managed by SecureConnector. The type of package you create is determined by the flavor of Linux operating system running on the targeted endpoints. Supported Linux package managers are as follows:

– Debian
– RPM

Installing SecureConnector as a Debian Package

SecureConnector is available for installation on Linux-based operating systems as a Debian package.

To work with Debian packages:

1. Log in to a CounterACT device and run the following command to create a 32-bit Debian package:

   ```
   fstool make_debian
   ```

   After the package is created, the following is displayed at the prompt:

   ```
   The package scon_<SecureConnector_version_number>_i386.deb is
   available at /usr/local/forescout/plugin/mac/debianpackage
   ```

   Do not change the name of the package.

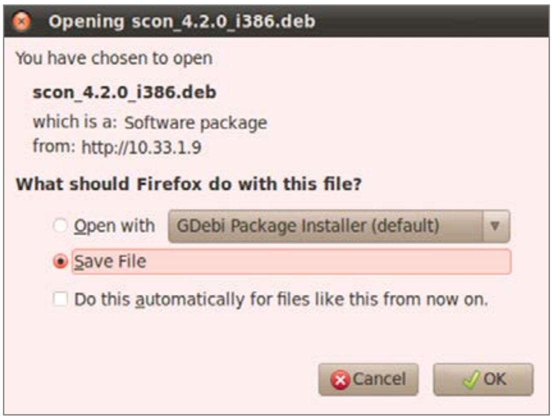
2. Copy the package to **`/usr/local/forescout/webapps/portal`**

3. Distribute the package to each endpoint, as follows:

   🗎 *Some commands may require root privileges.*

   🗎 *The package can be distributed and installed either by script or manually.*

   a. Download **`http://<appliance_ip>/name_of_package.deb.`**



Opening scon_4.2.0_i386.deb

You have chosen to open

scon_4.2.0_i386.deb

which is a: Software package
from: http://10.33.1.9

What should Firefox do with this file?

○ Open with GDebi Package Installer (default)
● Save File
□ Do this automatically for files like this from now on.

b. (for 64-bit Debian distributions) Submit the following command to accommodate the 32-bit Debian package:

```
dpkg --add-architecture i386
```

c. Install the file on the endpoint using the following command:

```
dpkg -i scon_<SecureConnector_version_number>_i386.deb
```

d. Start SecureConnector on the endpoint by running the following command:

```
/etc/init.d/scon start
```

e. (optional) You can also have SecureConnector start at boot time by adding it to the `rc.local` file or an equivalent method.

If necessary, you can remove the Debian package.

To remove the Debian package:

- On the endpoint, run the following command :

```
dpkg –r scon:i386
```

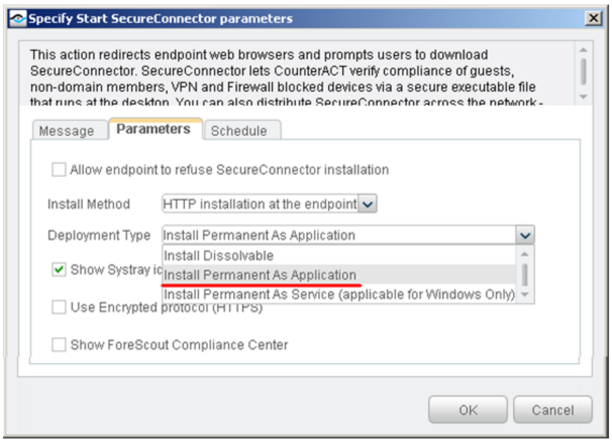
Installing SecureConnector as an RPM Package

SecureConnector is available for installation on Linux-based operating systems as an RPM Package Manager (RPM) package.

This installation is supported for CentOS, Fedora and Red Hat systems. In order to use this installation method, you must install SecureConnector on a *master machine*, create the RPM package on the master machine and then copy the RPM package to target machines.

To create the RPM package:

1. Install SecureConnector using the *Start SecureConnector* action on a master machine. In the action's **Parameters** tab, select the install method **HTTP installation at the endpoint** and the deployment type **Install Permanent as Application**.



Specify Start SecureConnector parameters

This action redirects endpoint web browsers and prompts users to download SecureConnector. SecureConnector lets CounterACT verify compliance of guests, non-domain members, VPN and Firewall blocked devices via a secure executable file that runs at the desktop. You can also distribute SecureConnector across the network -

□ Allow endpoint to refuse SecureConnector installation

Install Method: HTTP installation at the endpoint

Deployment Type: Install Permanent As Application

Install Dissolvable
Install Permanent As Application
Install Permanent As Service (applicable for Windows Only)

☑ Show Systray ic

□ Use Encrypted protocol (HTTPS)

□ Show ForeScout Compliance Center

2. Log in to an Appliance or Enterprise Manager and copy `/usr/local/forescout/plugin/mac/tools/sc2rpm.pl` to the master machine.

3. Run the following Perl script: `sc2rpm.pl --base=<SecureConnectorDir>`

where **<SecureConnectorDir>** is the directory where SecureConnector is installed. The default directory is **~/ForeScout/SecureConnector**.

4. The script generates an RPM file and saves it in a standard directory according to the Linux distribution. The RPM name format is:

   SecureConnector-<version>-<ip_address>-noarch.rpm

   Where **<version>** is the SecureConnector version and **<ip_address>** is the IP address of the CounterACT device assigned to manage the master machine. For example***:*** **SecureConnector-4.7-10_20_30_40-noarch.rpm**

5. Copy the RPM file to target machines and install it using a standard RPM command. This installs SecureConnector as a service (named *secureconnector*) and starts it using the **chkconfig** and **service** commands.

6. SecureConnector connects to the CounterACT Appliance to which the master machine is connected. If this is not the CounterACT Appliance assigned to manage the target machine, then SecureConnector should be redirected to the assigned Appliance using the Enterprise Manager.

Appendix 2: Linux Commands Used by the Plugin

This section lists Linux commands used by the Macintosh/Linux Property Scanner Plugin. Commands are used depending on the actions that are to be performed on the endpoint. This may affect the minimum privilege requirements for CounterACT as configured at the Appliance:

- If the plugin is used for monitoring and property resolution, regular privileges are sufficient.
- If remediation actions are used, you may need a higher privilege level.
- The plugin uses the sudo utility when super user access is required to run scripts on Linux endpoints, as when the *Run script as root user on endpoint* option is enabled for the **Run Script on Linux** action or the **Linux Expected Script Result** host property.

The following Linux commands are used to resolve properties and for actions by all inspection methods:

- **cat /etc/issue;uname –rs**: Operating system
- **hostname**
- **killall**: Process termination
- **ps -eo command c**: Processes
- **stat –t**: File-relevant properties
- **who**: Users

SecureConnector uses the following set of Linux commands:

- **awk**
- **cd**
- **chmod**
- **grep**
- **kill**
- **ln**
- **ls**
- **lsof –nP**
- **mv**
- **netstat –nlp**
- **nohup**
- **ps axwwo pid,ppid,command**
- **rm**
- **/sbin/runlevel**
- **ssh-keygen**
- **/usr/bin/ssh**

Appendix 3: Migrating Macintosh Endpoints Managed by SecureConnector to the OS X Plugin

The OS X Plugin now supports SecureConnector for endpoints running version 10.8 and above of the OS X operating system. The current release of the OS X plugin provides an updated version of SecureConnector native to OS X, and upgrades to this plugin will support future releases of the OS X operating system.

It is strongly recommended to install the OS X Plugin when you upgrade to this release of the Macintosh/Linux Property Scanner Plugin. The OS X Plugin installs the most recent version of SecureConnector on newly detected OS X devices, and supports the **Start/Stop SecureConnector** actions and other management functionality for these endpoints.

To maintain management continuity, this release of the Macintosh/Linux Property Scanner Plugin supports existing Macintosh endpoints using legacy versions of SecureConnector already installed on endpoints. It is strongly recommended to transfer existing endpoints running OS X 10.8 and above to the management control of the OS X Plugin, so they will receive updated versions of SecureConnector.

This section describes the upgrade and migration sequence in environments with existing OS X endpoints managed by SecureConnector.

To migrate Macintosh endpoints to the OS X Plugin:

1. Install this release of the Macintosh/Linux Property Scanner Plugin.
2. Install the latest release of the OS X Plugin.
3. Create a policy that uses the new **Macintosh SecureConnector Version** host property to detect existing OS X endpoints that run legacy versions of SecureConnector. The policy should apply the **Migrate to OS X SecureConnector** action to these endpoints.

   The OS X Plugin replaces the legacy version of SecureConnector on these endpoints with the latest version, and the endpoints now communicate with the OS X Plugin.

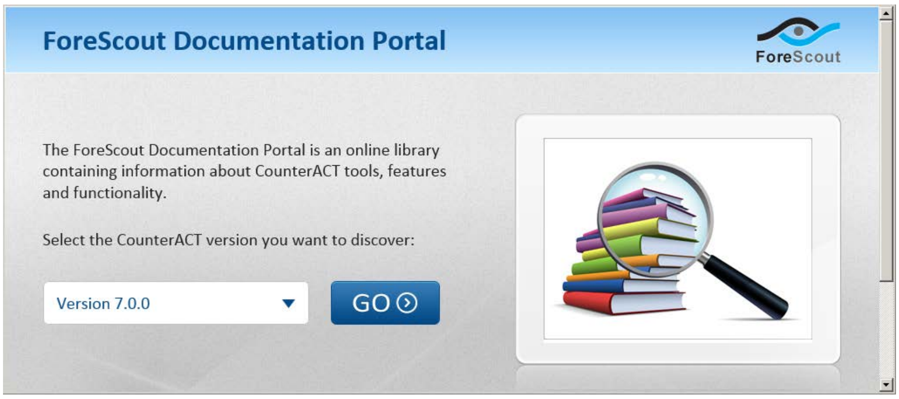
Additional CounterACT Documentation

For more detailed information about the CounterACT features described here or additional CounterACT features and plugins, refer to the following resources:

- Documentation Portal
- Customer Support Portal
- CounterACT Console Online Help Tools

Documentation Portal

The ForeScout Documentation Portal is a Web-based library containing information about CounterACT tools, features and functionality and integrations.



ForeScout Documentation Portal

The ForeScout Documentation Portal is an online library containing information about CounterACT tools, features and functionality.

Select the CounterACT version you want to discover:

To access the Documentation Portal:

1. Go to www.forescout.com/kb.
2. Use your customer support credentials to log in.
3. Select the CounterACT version you want to discover.

Customer Support Portal

The Customer Support Portal provides links to CounterACT version releases, service packs, plugins and modules as well as related documentation. The portal also provides a variety of How-to Guides, Installation Guides and more.

To access the Customer Support Portal:

1. Go to https://updates.forescout.com/support/index.php?url=counteract.
2. Select the CounterACT version you want to discover.

CounterACT Console Online Help Tools

Access information directly from the CounterACT Console.

Console Help Buttons

Use context sensitive *Help* buttons to quickly access information about the tasks and topics you are working with.

Console User Manual

1. Select **CounterACT Help** from the **Help** menu.

Plugin Help files

1. After the plugin is installed, select **Options** from the **Tools** menu and then select **Plugins**.
2. Select the plugin and then select **Help**.

Documentation Portal

1. Select **Documentation Portal** from the **Help** menu.

Legal Notice

These products are based on software developed by ForeScout. The products described in this document are protected by U.S. patents #6,363,489, #8,254,286, #8,590,004, #8,639,800 and #9,027,079 and may be protected by other U.S. patents and foreign patents.

Redistribution and use in source and binary forms are permitted, provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials and other materials related to such distribution and use acknowledge that the software was developed by ForeScout.

Unless there is a valid written agreement signed by you and ForeScout that governs the below ForeScout products and services:

- If you have purchased any ForeScout products, your use of such products is subject to your acceptance of the terms set forth at http://www.forescout.com/eula/;
- If you have purchased any ForeScout support service (“ActiveCare”), your use of ActiveCare is subject to your acceptance of the terms set forth at http://www.forescout.com/activecare-maintenance-and-support-policy/;
- If you have purchased any ForeScout Professional Services, the provision of such services is subject to your acceptance of the terms set forth at http://www.forescout.com/professional-services-agreement/;
- If you are evaluating ForeScout’s products, your evaluation is subject to your acceptance of the applicable terms set forth below:
  - If you have requested a General Availability Product, the terms applicable to your use of such product are set forth at: http://www.forescout.com/evaluation-license/.
  - If you have requested an Early Availability Product, the terms applicable to your use of such product are set forth at: http://www.forescout.com/early-availability-agreement/.
  - If you have requested a Beta Product, the terms applicable to your use of such product are set forth at: http://www.forescout.com/beta-test-agreement/.
  - If you have purchased any ForeScout Not For Resale licenses, such license is subject to your acceptance of the terms set forth at http://www.forescout.com/nfr-license/.

Send comments and questions about this document to: documentation@forescout.com

2016-06-30 18:16